



Building Programs
Technical Upgrade

Operating System/2
Programming Tools
and Information
Version 1.2/1.3

91F8682



Building Programs
Technical Upgrade

Operating System/2
Programming Tools
and Information
Version 1.2/1.3

First Edition (December 1990)

The following paragraph does not apply to the United Kingdom or any country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This publication could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time.

It is possible that this publication may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country.

Requests for technical information about IBM products should be made to your IBM Authorized Dealer or your IBM Marketing Representative.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Commercial Relations, IBM Corporation, Purchase, NY 10577.

COPYRIGHT LICENSE: This publication contains printed sample application programs in source language, which illustrate OS/2 programming techniques. You may copy and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the OS/2 application programming interface.

Each copy of any portion of these sample programs or any derivative work, which is distributed to others, must include a copyright notice as follows: "© (your company name) (year) All Rights Reserved."

© Copyright International Business Machines Corporation 1990. All rights reserved.

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Special Notices

The following terms, denoted by an asterisk (*) in this publication, are trademarks of the IBM Corporation in the United States and/or other countries:

IBM
IBM C/2
IBM COBOL/2
IBM FORTRAN/2
IBM Macro Assembler/2
Operating System/2
OS/2
Presentation Manager
Systems Application Architecture.

The following terms, denoted by a double asterisk (**) in this publication, are trademarks of other companies as follows:

Microsoft Corporation

About This Book

This book, as well as the other books in the upgrade package, contains programming information that is new to or different from the information in the *IBM Operating System/2® (OS/2®) Programming Tools and Information Version 1.2* library (part number 6024929). In addition, information from the *OS/2 Programming Tools and Information Version 1.2 Technical Update* (part number 64F1705) is included in this upgrade package.

The upgrade describes features added by the OS/2 Version 1.3 product, and amends some of the information that was published with OS/2 Version 1.2.

This book is a companion to *OS/2 Version 1.2 Building Programs*. **Only those chapters of the Version 1.2 book that are changed by the Version 1.3 product are included herein.** The chapter numbers and titles are the same as those in the Version 1.2 book. As a convenience, the beginning of each chapter in this book provides a summary of changes for that chapter.

Who Should Read This Book

This book is for application designers and programmers who are using the components of the *IBM OS/2 Version 1.2/1.3 Programming Tools and Information Technical Upgrade* package. The reader is assumed to be familiar with the services of OS/2.

* Trademark of the IBM Corporation

Contents

Chapter 2. Building a Presentation Manager Application	1
Resource Script File	1
Chapter 7. LINK Options	3
Summary of LINK Options	3
/? Viewing the Options List	4
/ALIGNMENT Aligning Segments	4
/BATCH Running in Batch Mode	5
/CODEVIEW Preparing Files for CodeView	5
/CPARMAXALLOC Reserving Paragraph Space	6
/DOSSEG Ordering Segments	7
/DSALLOCATE Controlling Data Loading	7
/EXEPACK Packing Executable Files	8
/FARCALLTRANSLATION Optimizing Translation of Far Calls	8
/HELP Viewing the Options List	9
/HIGH Controlling where the Executable File is Loaded	9
/INFORMATION Displaying Information about the Linking Process	10
/KNOWEAS Knowing an Application Has Extended Attributes	10
/LINENUMBERS Copying Line Numbers to the Map File	11
/MAP Producing a Public Symbol Map	11
/NODEFAULTLIBRARYSEARCH Ignoring Default Libraries	12
/NOEXTDICTIONARY Preventing Extended Dictionary Search	12
/NOFARCALLTRANSLATION Disabling Far-Call Translations	13
/NOGROUPASSOCIATION Maintaining Compatibility	13
/NOIGNORECASE Recognizing Uppercase and Lowercase	14
/NOLOGO Disable SignOn Banner	14
/NONULLSDOSSEG Ordering Segments without Nulls	15
/NOPACKCODE Disabling Code-Segment Packing	15
/OVERLAYINTERRUPT Overriding the DOS Interrupt Number	16
/PACKCODE Packing Code Segments	16
/PACKDATA Packing Data Segments	17
/PAUSE Pausing to Change Disks	17
/PMTYPE Naming the Application Type	18
/SEGMENTS Setting the Maximum Number of Segments	19
/STACK Setting the Stack Size	19
/TINY Generating the .COM File	20
/WARNFIXUP Warning of Incorrect Offset	21
Using LINK Options	21
Entry of Numeric Parameters	21
Chapter 8. Module Definition File Statements	23
NEWFILES Supporting Non-8.3 File Names	23
Index	25

Chapter 2. Building a Presentation Manager Application

Summary of Changes

New	Updated	Section Title
	√	Resource Script File

Resource Script File

On page 2-7, under **MENU ID_TEMPLATE**, add the following note:

Note: When defining a menu action containing a mnemonic character, select only those characters that have an uppercase equivalent. For example, the character *y umlaut* does not have an uppercase equivalent. Thus, the character should not be used for mnemonic representation. A conflict may occur when international characters do not have uppercase equivalents.

Chapter 7. LINK Options

Summary of Changes

New	Updated	Section Title
✓		/?
✓		/BATCH
	✓	/EXEPACK
✓		/KNOWEAS
	✓	/NOEXTDICTIONARY
✓		/NOLOGO
✓		/NONULLSDOSSEG
✓		/PMTYPE
✓		/TINY

This section completely replaces Chapter 7.

Summary of LINK Options

The LINK options are:

<i>Option</i>	<i>Description</i>
/?	Writes a list of the available options to the screen.
/ALIGNMENT	Sets segment alignment factor.
/BATCH	Disables the default prompt that requests a new path name.
/CODEVIEW	Includes debugging information for the CodeView debugger.
/CPARMAXALLOC	Changes value of maximum number of reserved paragraphs.
/DOSSEG	Forces ordering of segments.
/DSALLOCATE	Controls data loading.
/EXEPACK	Packs executable files.
/FARCALLTRANSLATION	Optimizes intersegment far calls.
/HELP	Writes a list of the available options to the screen.
/HIGH	Controls loading of the run file.
/INFORMATION	Displays information about the linking process.
/KNOWEAS	Tells LINK the application has extended attributes.
/LINENUMBERS	Copies line numbers to the map file.
/MAP	Lists all public symbols in your program.
/NODEFAULTLIBRARYSEARCH	Ignores default libraries.
/NOEXTDICTIONARY	Prevents LINK from searching the extended dictionary.
/NOFARCALLTRANSLATION	Disables far-call translations.

/NOGROUPASSOCIATION	Provides compatibility with previous compiler versions.
/NOIGNORECASE	Differentiates between uppercase and lowercase letters.
/NOLOGO	Disables the signon banner displayed when LINK starts.
/NONULLSDOSSEG	Forces ordering of segments without nulls.
/NOPACKCODE	Disables code-segment packing.
/OVERLAYINTERRUPT	Overrides the DOS interrupt number.
/PACKCODE	Packs code segments.
/PACKDATA	Packs data segments.
/PAUSE	Pauses the link process so diskettes can be changed.
/PMTYPE	Specifies the type of application being linked.
/SEGMENTS	Sets the maximum number of segments.
/STACK	Sets the stack size.
/TINY	Generates a .COM output file rather than a .EXE file.
/WARNFIXUP	Warns of incorrect offset.

/?

Viewing the Options List

Purpose

To write a list of the available options to the screen.

Format

/?

Remarks

- **/?** option and **/HELP** option provide the same information.
- Do not give a file name when using the **/?** option.

/ALIGNMENT

Aligning Segments

Purpose

To set the segment alignment factor in the executable file to a specified number of bytes.

Format

/ALIGNMENT:*number*

The minimum abbreviation is **/A**. [*number* can be a decimal, hexadecimal, or octal number.]

Defaults

The default alignment factor is 512.

Restrictions

- *number* must be a power of 2.
- This option is valid only for code that is linked to run in the OS/2 environment.

Remarks

LINK aligns segments to ensure proper loading. Aligning a segment means moving its address to the next address boundary divisible by the alignment factor.

We recommend an alignment factor of 16 for Presentation Manager applications.

/BATCH

Running in Batch Mode

Purpose

To disable the default prompt generated by LINK. LINK prompts you for a new path name whenever it cannot find an object file or library it was directed to use. With the /BATCH, LINK generates an appropriate error or warning message and leaves the external reference unresolved.

Format

/BATCH

The minimum abbreviation is **/BA**.

Remarks

The primary use of this option is to call LINK from a batch file or a *Make* utility.

Note that this option does not affect prompts for command-line input.

/CODEVIEW

Preparing Files for CodeView

Purpose

To include symbolic debugging information for CodeView** in the executable file.

** Trademark of Microsoft Corporation

Format

/CODEVIEW

The minimum abbreviation is **/CO**.

Restrictions

/CODEVIEW cannot be used with /EXEPACK.

/CPARMAXALLOC

Reserving Paragraph Space

Purpose

To change the default value of the MAXALLOC field.

Format

/CPARMAXALLOC: *number*

The minimum abbreviation is **/CP**.

Defaults

The default for the MAXALLOC field is 65535 (decimal), that is, 64K minus 1.

Restrictions

- This option is valid only for code linked to run in the DOS environment or in the DOS mode of OS/2.
- The maximum number of paragraphs reserved for an application is determined by the value of the MAXALLOC field at offset 0x0C in the .EXE header.
- If the value you specify is less than the computed value of MINALLOC (at offset 0x0A), LINK uses the value of MINALLOC instead.

Remarks

A paragraph is the smallest storage unit (16 bytes) addressable by a segment register. This field controls the maximum number of paragraphs reserved in memory for your application.

You can reset the default to any number between 1 and 65535 (decimal) using decimal, hexadecimal, or octal notation. Changing the default is helpful because reserving all available memory does not increase the performance of your application.

If your application invokes another application, you need to reserve memory for that application.

/DOSSEG

Ordering Segments

Purpose

To order segments in the following sequence:

- All segments having a class name ending in CODE
- All other segments outside of DGROUP
- DGROUP segments in the following order:
 - Any segments of class BEGDATA reserved
 - Any segments not of class BEGDATA, BSS, or STACK
 - Segments of class BSS
 - Segments of class STACK.

Format

/DOSSEG

The minimum abbreviation is **/DO**.

Remarks

Linking with the standard C-2 run-time libraries automatically enables the /DOSSEG option in the startup module. Use the /NONULLSDOSSEG option to override the /DOSSEG comment record.

Using this option inserts 16 bytes of nulls in front of the segment named `_TEXT`, which, if present, is a requirement for C/2 run-time support.

/DSALLOCATE

Controlling Data Loading

Purpose

To load all data starting at the high end of the data segment.

Format

/DSALLOCATE

The minimum abbreviation is **/DS**.

Defaults

By default, LINK loads all data starting at the low end of the data segment. At run time, LINK sets the data segment pointer to the lowest possible address to allow the entire data segment to be used.

Restrictions

This option is valid only for code linked to run in the DOS environment or in the DOS mode of OS/2.

Remarks

Set the data segment pointer at run time to the lowest data segment address containing program data.

The /DSALLOCATE option typically is used with the /HIGH option to take advantage of unused memory within a data segment. You can reserve available memory below the area specifically reserved for DGROUP by using the same data-segment pointer.

/EXEPACK

Packing Executable Files

Purpose

To remove sequences of repeated bytes (typically nulls) and to make most efficient use of the load-time relocation table before creating a DOS executable file.

Format

/EXEPACK

The minimum abbreviation is **/E**.

Restrictions

- This option is valid only for code linked to run in the DOS environment or in the DOS mode of OS/2.
- /EXEPACK cannot be used with /CODEVIEW, because symbolic debuggers cannot be used with packed files.

Remarks

OS/2 executable files are compressed only if /EXEPACK is specified.

The /EXEPACK option does not always save disk space and may sometimes increase file size. Applications having a large number of load-time relocations (500 or more), or long streams of repeated characters, are usually shorter if packed.

Note: The LINK options /EXEPACK and /HIGH are mutually exclusive.

/FARCALLTRANSLATION

Optimizing Translation of Far Calls

Purpose

To optimize translation of intersegment far calls using the sequence:

```
NOP  
PUSH CS  
CALL NEAR address
```

Format

/FARCALLTRANSLATION

The minimum abbreviation is **/FAR**.

Defaults

The default option is **/NOFARCALLTRANSLATION**.

Restrictions

This option is valid only for code linked to run in the OS/2 environment.

In translating far calls, LINK may mistake a byte that uses the value 0x9A as a constant for a far call.

Remarks

Using this option for most medium and large model applications makes the executable file smaller, thus saving load time.

/HELP

Viewing the Options List

Purpose

To write a list of the available options to the screen.

Format

/HELP

The minimum abbreviation is **/HE**.

Remarks

Do not give a file name when using the **/HELP** option.

/HIGH

Controlling where the Executable File is Loaded

Purpose

To load the executable file as high as possible in memory without overlaying the transient portion of COMMAND.COM.

Format

/HIGH

The minimum abbreviation is **/HI**.

Defaults

COMMAND.COM is loaded at the highest area of memory; the executable file, at the lowest end.

Restrictions

- This option is valid only for code linked to run in the DOS environment or in the DOS mode of OS/2.
- The /HIGH option should not be used with programs written in C Language.

Remarks

Use the /HIGH option in association with the /DSALLOCATE option.

Note: The LINK options /EXEPACK and /HIGH are mutually exclusive.

/INFORMATION

Displaying Information about the Linking Process

Purpose

To display information, such as the names of files being linked during the linking process.

Format

/INFORMATION

The minimum abbreviation is /I.

Remarks

/INFORMATION is useful for debugging.

/KNOWEAS

Knowing an Application Has Extended Attributes

Purpose

To tell LINK that the application has extended attributes. Use KNOWEAS when linking a real-mode application.

Format

/KNOWEAS

The minimum abbreviation is /K.

/LINENUMBERS

Copying Line Numbers to the Map File

Purpose

To copy the line number of the starting address of each instruction in the application source code.

Format

/LINENUMBERS

The minimum abbreviation is **/LI**.

Restrictions

The numbering of lines of source code is supported by most high-level languages, but not Macro Assembler. To obtain the line numbers associated with the starting address of each instruction, you must request that LINK generate a map file.

Remarks

If an object file has no line-number information, LINK ignores the /LINENUMBERS option.

If you do not specify a map file in a LINK command, you can still use the /LINENUMBERS option to create a map file. Place the option at or before the List File [NUL.MAP]: prompt. LINK gives the map file the same file name as the first object file specified in the command and gives it the default .MAP file extension.

/MAP

Producing a Public Symbol Map

Purpose

To produce a listing of all the public symbols declared in your application. This list is copied to the map file created by LINK.

Format

/MAP[:*number*]

The minimum abbreviation is **/M**. *number* can be a decimal, hexadecimal, or octal number.

Defaults

If you do not specify a value for *number*, the default is 2048.

Restrictions

Valid values are 1 through 32767.

Remarks

The *number* parameter specifies the maximum number of public symbols that LINK can sort in the map file.

Specifying a number generates a map file with the public symbols sorted by address rather than by name. If you do not need the list of public symbols sorted by name, you can tell LINK not to include it. When you specify the /MAP option, include a number large enough to accommodate only the number of public symbols in the application.

If you do not request a map file in a LINK command, you can use the /MAP option to create a map file. LINK gives the map file the same name as the first object file specified in the command and the default .MAP file extension.

For a description and examples of the format of a map file, see "LINK Output as a Debugging Aid" on page 6-5 of the original *Building Programs* book.

/NODEFAULTLIBRARYSEARCH **Ignoring Default Libraries**

Purpose

To tell LINK to ignore any default libraries named in object files.

Format

/NODEFAULTLIBRARYSEARCH

The minimum abbreviation is **/NOD**.

Remarks

High-level language compilers sometimes include library names in the object files they generate. Using this option ensures that only the libraries you name are linked with your object files.

/NOEXTDICTIONARY **Preventing Extended Dictionary Search**

Purpose

To prevent LINK from searching the extended dictionary. The extended dictionary is a list of inter-library dependencies. For example, suppose *FOO*, *X*, and *Y* are procedure names; after LINK finds *FOO*, it knows from the extended dictionary that *X* and *Y* are required from the same library.

Format

/NOEXTDICTIONARY

The minimum abbreviation is **/NOE**.

Defaults

The extended dictionary is searched by default.

Restrictions

None.

Remarks

The extended dictionary quickens library searches, so, in general, /NOE causes linking to take longer. A library without an extended dictionary is still a valid library.

/NOE also is used to redefine a symbol that is already defined in a library. In this case, if you fail to use the /NOEXTDICTIONARY switch, a *symbol multiply defined* error occurs.

/NOFARCALLTRANSLATION

Disabling Far-Call Translations

Purpose

To disable translation of intersegment far calls.

Format

/NOFARCALLTRANSLATION

The minimum abbreviation is **/NOF**.

Defaults

This is the default (see /FARCALLTRANSLATION).

Restrictions

This option is valid only for the OS/2 environment.

/NOGROUPOSSOCIATION

Maintaining Compatibility

Purpose

To process a specific set of fix-up routines for compatibility with earlier versions of LINK and language compilers.

Format

/NOGROUPOSSOCIATION

The minimum abbreviation is **/NOG**.

Restrictions

- This option is valid only for code linked to run in the DOS environment or in the DOS mode of OS/2.
- It should not be used with applications written in C Language.

Remarks

None.

/NOIGNORECASE Recognizing Uppercase and Lowercase

Purpose

To recognize uppercase and lowercase letters as distinct characters when used in symbol names.

Format

/NOIGNORECASE

The minimum abbreviation is **/NOI**.

Examples

LINK recognizes uppercase and lowercase letters as identical. For example, it considers *TWO*, *Two*, and *two* the same. Using this option tells LINK to recognize the three symbol names as distinct.

Remarks

This option is typically used with object files created by high-level language compilers. Some compilers recognize uppercase and lowercase letters as distinct letters and assume that LINK does the same.

/NOLOGO Disable SignOn Banner

Purpose

To disable the signon banner displayed when LINK starts.

Format

/NOLOGO

The minimum abbreviation is **/NOL**.

/NONULLSDOSSEG

Ordering Segments without Nulls

Purpose

To arrange segments in a certain sequence. This option is the same as the /DOSSEG option, except that /NONULLSDOSSEG does not insert 16 bytes of nulls in front of the segment named `_TEXT` (if the `_TEXT` segment is defined).

Format

/NONULLSDOSSEG

The minimum abbreviation is **/NON**.

Remarks

Linking with the standard C-2 run-time libraries automatically enables the /DOSSEG option in the startup module. Use the /NONULLSDOSSEG option to override the /DOSSEG comment record.

/NOPACKCODE

Disabling Code-Segment Packing

Purpose

To tell LINK not to pack neighboring logical code segments into one physical segment.

Format

/NOPACKCODE

The minimum abbreviation is **/NOP**.

Defaults

The default option is /PACKCODE.

Restrictions

This option is valid only for code linked to run in the OS/2 environment.

Remarks

For more information about packing, see the /PACKCODE option.

/OVERLAYINTERRUPT

Overriding the DOS Interrupt Number

Purpose

To override the default DOS interrupt number for passing control to overlays.

Format

/OVERLAYINTERRUPT:*number*

The minimum abbreviation is **/O**. *number* can be a decimal number from 0 to 255, an octal number from 0 to 0377, or a hexadecimal number from 0 to 0xFF.

Defaults

0x3F is the default DOS interrupt number.

Restrictions

This option is valid only for code linked to run in the DOS environment or in the DOS mode of OS/2.

Remarks

Numbers that conflict with DOS interrupts are not prohibited, but we do not recommend using them.

/PACKCODE

Packing Code Segments

Purpose

To tell LINK to pack neighboring logical code segments into one physical segment.

Format

/PACKCODE[*:packlimit*]

The minimum abbreviation is **/PACKC**. *packlimit* is the maximum size of the segment, in bytes.

Defaults

- This option is the default. **/NOPACKCODE** should be used to override **/PACKCODE**.
- If **/PACKCODE** is entered without a *packlimit*, LINK uses the default of 65536.

Restrictions

This option is valid only for code linked to run in the OS/2 environment.

Remarks

The optional *packlimit* is the limit at which to stop packing.

Note: Code-segment lengths of 65501 – 65536 may be unreliable for the 80286 microprocessor. Use of these segments will generate the following warning message:

LINK: warning L4011:
PACKCODE value exceeding 65500 unreliable.

/PACKDATA

Packing Data Segments

Purpose

Tells LINK to pack neighboring logical data segments into one physical segment.

Format

/PACKDATA[:*packlimit*]

The minimum abbreviation is **/PACKD**.

Defaults

By default, LINK does not try to pack neighboring logical data segments into one physical segment. *packlimit* is the limit at which to stop packing. If no number is given, LINK uses 65536.

Restrictions

This option is valid for OS/2 executable programs only.

Remarks

Consider using this option if you have a large-model application containing many files and you receive the following message: L1073-file-segment limit exceeded.

/PAUSE

Pausing to Change Disks

Purpose

To tell LINK to pause before writing the executable file to a disk file, so you can change diskettes.

Format

/PAUSE

The minimum abbreviation is **/PAU**.

Remarks

If you use the /PAUSE option, LINK displays the following message before creating the executable file:

```
About to generate .EXE file
Change diskette in drive letter and press Enter
```

where *letter* is the correct drive name. This message appears after LINK has read data from the object and library files, and after it has written data to a map file. LINK resumes processing when you press Enter.

After LINK writes the executable file to a disk file, the following message appears:

```
Please replace original diskette in drive letter and press Enter
```

Note: Do not remove the diskette containing the temporary file. If the following message appears, press Ctrl and Break keys to end the LINK session.

```
LINK: fatal error L1087:
unexpected end-of-file on scratch file
```

Rearrange your files so LINK can write the temporary file and the executable file to the same diskette and try again.

/PMTYPE

Naming the Application Type

Purpose

To specify the type of application being linked.

Format

```
/PMTYPE[:type]
```

The minimum abbreviation is **/PM**.

Restrictions

This option is valid for OS/2 executable files only.

Remarks

The /PMTYPE option is equivalent to the NAME statement in a module definition file. A keyword in *type* is the same as a keyword in a NAME statement, as demonstrated by the following:

Type Field Keyword	Equivalent Keyword in NAME Statement
PM	WINDOWAPI
VIO	WINDOWCOMPAT
NOVIO	NOTWINDOWCOMPAT

/SEGMENTS

Setting the Maximum Number of Segments

Purpose

To process no more than a specified number of segments for each application.

Format

/SEGMENTS:*number*

The minimum abbreviation is **/SE**. *number* can be any integer value in the range of 1 to 3072.

Defaults

- The **/SEGMENTS** option bypasses the default limit of 128 segments.
- If you do not specify **/SEGMENTS**, LINK reserves enough memory to process up to 128 segments. If your program has more than 128 segments, you must set the segment limit higher to increase the number of segments that LINK can process.

Examples

This example sets the segment limit to 192:

```
LINK file/SE:192,,;
```

This example sets the segment limit to 255 (0xFF):

```
LINK moda+modb,run/SEGMENTS:0xff,ab,em+mlibfp;
```

Remarks

Set the segment limit lower if you get the following LINK error message:

```
LINK: error L1054:  
requested segment limit too high.
```

/STACK

Setting the Stack Size

Purpose

To set the application stack to a specified number of bytes.

Format

/STACK:*size*

The minimum abbreviation is **/ST**. *size* can be any positive integer value in the range of 0 to 65534.

Examples

This example sets the stack size to 512 bytes:

```
LINK file/STACK:512,,;
```

This example sets the stack size to 255 (0xFF) bytes:

```
LINK moda+modb,run/ST:0xFF,ab,\lib\start;
```

This example sets the stack size to 24 (030 octal) bytes:

```
LINK startup+file/ST:030,,;
```

Remarks

LINK automatically calculates the size of the stack allocated to an application. The stack size is based on the size of the stack segments specified in the object files. If you specify the /STACK option, LINK uses the specified *size* in place of the value it calculated.

The stack size also can be changed with the STACKSIZE statement in the module definition file.

/TINY

Generating the .COM File

Purpose

To create a file that defaults to an extension of .COM rather than .EXE.

Format

```
/TINY
```

The minimum abbreviation is /T.

Restrictions

- Only one physical segment is permitted with each program. You can declare more than one segment in an assembly-language program, but the segments must be in the same group.
- Far-segment references are not permitted in the program. This means segment addresses cannot be used as immediate data for instructions. For example, the following instruction is not allowed:

```
mov    ax,CODESEG
```

- Programs for OS/2 cannot be linked to the .COM file. .COM files are valid only in real mode.

Remarks

Using this option, LINK automatically sets the /FARCALLTRANSLATION option to optimize far calls. This increases your chances of generating a .COM file format.

Note: LINK creates a separate file with debugger information when the /CODEVIEW option is used with the /TINY option. LINK supplies the same file name as the base file name, but with a .DBG extension.

/WARNFIXUP

Warning of Incorrect Offset

Purpose

To issue a warning for each segment relative fix-up of location-type offset. This option should be used specifically if the segment is contained within a group, but is not at the beginning of the group.

Format

/WARNFIXUP

The minimum abbreviation is **/W**.

Remarks

LINK includes the displacement of the segment from the group in determining the final value of the fix-up. This is the opposite of what happens with DOS executable files.

Note: Valid for OS/2 executable files only.

Using LINK Options

When using LINK options, follow these guidelines.

- You can specify an option on any line before the last comma at the end of the LINK command or response line.
- Every option must begin with the slash (/) character, even if other options appear before it on the line.
- You can abbreviate option names as long as the abbreviations contain enough letters to distinguish the specified option from other options. Minimum abbreviations are listed with the description for each option.
- LINK does not recognize spaces between characters, nor transposed letters.

Entry of Numeric Parameters

Numeric parameters in LINK options can be entered in decimal, hexadecimal, or octal. The format follows C Language conventions, which are:

Decimal	Any number that begins with anything other than zero. For example: 1, 65536, 2084, 234.
Hexadecimal	Any number that begins with 0x and contains the digits 0 through 9, and the letters A through F. For example: 0xFFFF, 0x10.
Octal	Any number that begins with the digit 0 and contains only the digits 0 through 7. For example: 010, 05000, 0777.

Chapter 8. Module Definition File Statements

Summary of Changes

New	Updated	Section Title
	√	NEWFILES

NEWFILES

Supporting Non-8.3 File Names

On page 8-8, change the **Format** parameter to the following:

NAME *programname* NEWFILES

In addition, change the section under **Examples** to the following:

NAME *programname* NEWFILES

Index

C

CodeView, debugging with 5

D

data segment pointer 7
data segments, packing 16
debugging with CodeView 5
DGROUP 7
diskette changing 17

E

executable files 9
executable files, packing 8

F

far call translations 8, 13

I

interrupt, setting 16

L

libraries, ignoring defaults 12
line numbers 11
LINK compatibility with early versions 13
LINK options
 summary 3
 using numbers 21
 /ALIGNMENT 4
 /BATCH 5
 /CODEVIEW 5, 8
 /CPARMAXALLOC 6
 /DOSSEG 7
 /DSALLOCATE 7
 /EXEPACK 8
 /FARCALLTRANSLATION 8
 /HELP 9
 /HIGH 8, 9
 /INFORMATION 10
 /KNOWEAS 10
 /LINENUMBERS 11
 /MAP 11
 /NODEFAULTLIBSEARCH 12
 /NOEXTDICTIONARY 12
 /NOFARCALLTRANSLATION 13
 /NOGROUPASSOCIATION 13
 /NOIGNORECASE 14
 /NOLOGO 14
 /NONULLSDOSSEG 15
 /NOPACKCODE 15

LINK options (*continued*)

/OVERLAYINTERRUPT 16
/PACKCODE 16
/PACKDATA 17
/PAUSE 17
/PMTYPE 18
/SEGMENTS 19
/STACK 19
/TINY 20
/WARNFIXUP 21
/? 4
LINK, using 21
loading executable files 9
lowercase use with LINK 14

M

MAXALLOC field 6
MINALLOC field 6

N

NEWFILES statement 23

P

packing data segments 17
packing executable files 8
paragraph space, reserving 6
pause before writing to disk 17
pointer, data segment 7
preparing files for CodeView 5

R

references, unresolved 21
reserving paragraph space 6
run file loading 9

S

segment limit 19
segment offset 21
segment pointer 7
setting, interrupts 16
stack size, setting 19

U

unresolved references 21
uppercase use with LINK 14

IBM United Kingdom
International Products Limited
PO Box 41, North Harbour
Portsmouth, PO6 3AU
England

Printed in Denmark by
J. H. Schultz Print a/s
Copenhagen

